

AMENDMENTS TO THE SPECIFICATION

Please replace paragraphs [0025], [0026], [0029] and [0051] of the specification with the following amended paragraphs, shown below with mark-up:

[0025] Table I below provides an example set of object class definitions contained in [[MDS]] MDA schema definition files 124 for the “Interfaces”; “InterfaceA”; “InterfaceB” and “InterfacesCommon” classes from the prior example. Each class definition includes a class name, a parent class, a class category (container or leaf), a description and a pathname. The parent class is used to construct the trees and the pathname is used to define the location of the data on system DB 108. In the present example, the Interfaces class is a child container node of the RootCfg node, which is also depicted in FIG. 2. The InterfaceA, InterfaceB and InterfaceCommon classes are child leaf nodes of the Interfaces container node.

[0026] Table II below table provides an example of two other leaf classes defined by [[MDS]] MDA schema definition files 124. Each leaf class definition specifies a class name, a parent class, a class category (leaf), a description, a pathname a value and a default value. The second leaf class "BGPDampening" includes five data types that together define the BGP dampening characteristics for this class.

[0029] During the second step of the MDA design phase 122, MDA schema definition files 124 are processed by an MDA schema compiler 126 that generates both MDA schemas 128 and XML schemas 130. According to one embodiment of the invention, MDA schema compiler 126 generates an internal version of the MDA schema definition files 124 in the form of a parsed tree representation. The MDA schemas 128 and XML schemas 130 are then generated based upon the internal version of the MDA schema definition files 124. According to one embodiment of the invention, MDA schemas 128 are binary versions of [[MDS]] MDA schema definition files

124 suitable for loading onto router 102, and are used by router 102 at runtime to process XML requests received from client 104. XML schemas 130 are text files that conform to the W3C schema standard for defining XML content. XML schemas 130 document for developers the external XML interface to router 102. XML schemas 130 are also used during the client design phase as described in more detail hereinafter.

[0051] Server side processing is now described with reference to a flow diagram 600 of FIG. 6. FIG. 6 depicts high level server side processing in the context of router 102 processing an XML request from client 104. In step 602, router 102 receives an XML request from client 104. In step 604, the XML request is parsed to identify XML elements contained in the request. For example, the XML request may be examined to identify XML tags contained therein. In step 606, a relationship is determined between each XML element and data objects defined by MDA schemas 128. This may include, for example, identifying management data items maintained in system DB 108 by router 102 that correspond to the XML tags contained in the XML request. In step 608, an XML tree is built and one or more data requests generated and provided to [[MDS]] MDA schema server 106. The data requests specify operations to be performed on the identified data items that correspond to the XML tags. In step 610, the data requests are processed against system DB 108. In step 612, an XML response is generated by router 102 and provided to client 104. The XML response may contain management data requested by client 104. The XML response may instead contain a confirmation that the requested operation has been performed. For example, the XML response may contain a confirmation that management data has been updated on router 102. As another example, the XML response may contain a confirmation that updated management data has been implemented by router 102. This may be provided, for example, in response to a request to commit changes to management data on router 102.